CS 491/591 Spring 2013 Lab 2 – 10 points

This is due as a physical turn-in at the beginning of class on Tuesday, 12 February. You will turn in a single sheet of 8.5” by 11” paper with writing/drawing on only one side. The sheet should not be stapled or folded. It should have your name on it. Your name, paragraph, and drawing should all be on one side of the paper, and the other side completely blank.

You may discuss this assignment with your classmates and others at a high level, but you should do your own setup work and own programming. Your solution to the assignment must be unique. You can use any online or offline resources you wish, as long as you do your own assignment.

The purpose of this assignment is to understand IP fragmentation and explore how expressive the different views of a re-constructed IP packet can be.

You’ve already been given Python source code for checking what reassembly policy a machine uses. You should build a private network of four virtual machines (or actual machines, if you prefer): Ubuntu Linux 12.04, FreeBSD 9.1, something that uses the “First” policy, and then a machine for sending UDP packets (perhaps another Linux machine). For OSes that use the “First” policy, I recommend OpenIndiana or Solaris, but Windows and MacOS might also work. If you want to also dig up a Cisco router and HP JetDirect printer to cover all five reassembly policies, that would certainly be worth some brownie points.

Your goal is to do a demo with real packets that demonstrates the expressiveness of IP fragment reassembly in a creative way. For example, you might send the same sequence of UDP packet fragments to each of the three recipient machines and have each machine see a different Shakespeare quote. Or, you might attack the Windows machine and have Wireshark on the Linux and BSD machines see two distinctly different things that are not Windows attacks, nor what the other UNIX-like OS saw. You can set it up so that, e.g., the Windows machine is an endpoint and the packet gets routed through (and captured by) the Linux and BSD machines, for example. Or you could send a buffer overflow exploit that works for whatever OS receives it because of how the fragments get reassembled.

Your color drawing should illustrate how the same information is interpreted in more than one way, or how the different policies mask out the other parts that you don't want that particular machine to see. Again, be creative, don't give me a technical diagram of packet fragments being reassembled. Your goal is to inspire your classmates by showing them something they couldn't have imagined was possible with IP fragmentation. Your picture should capture the awesomeness of what you did.